[JAVA调用Shell脚本](http://blog.csdn.net/jj12345jj198999/article/details/11891701)

分类： [Felven在职场](http://blog.csdn.net/jj12345jj198999/article/category/926786)2013-09-22 11:03 4940人阅读 [评论](http://blog.csdn.net/zhaoxinfan/article/details/11891701#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/zhaoxinfan/article/details/11891701#report)

[shell](http://www.csdn.net/tag/shell) [java](http://www.csdn.net/tag/java" \t "_blank)

在实际项目中，JAVA有时候需要调用C写出来的东西，除了JNI以外，我认为一种比较好的方法是JAVA调用Shell。先把C写出来的make成可执行文件，然后再写一个shell脚本执行该可执行文件，最后是JAVA调用该shell脚本。

JAVA调用很简单，例子如下：

首先是shell脚本

**[plain]** [view plaincopyprint?](http://blog.csdn.net/zhaoxinfan/article/details/11891701)

1. #!/bin/sh
2. echo Begin word cluster
3. /home/felven/word2vec/word2vec -train /home/felven/word2vec/resultbig.txt -output /home/felven/word2vec/classes.csv -cbow 0 -size 200 -window 5 -negative 0 -hs 1 -sample 1e-3 -threads 12 -classes 2000
4. echo The word classes were saved to file classes.csv

然后是JAVA调用代码

**[java]** [view plaincopyprint?](http://blog.csdn.net/zhaoxinfan/article/details/11891701)

1. **import** java.io.BufferedReader;
2. **import** java.io.InputStreamReader;
4. **public** **class** RunShell {
5. **public** **static** **void** main(String[] args){
6. **try** {
7. String shpath="/home/felven/word2vec/demo-classes.sh";
8. Process ps = Runtime.getRuntime().exec(shpath);
9. ps.waitFor();
11. BufferedReader br = **new** BufferedReader(**new** InputStreamReader(ps.getInputStream()));
12. StringBuffer sb = **new** StringBuffer();
13. String line;
14. **while** ((line = br.readLine()) != **null**) {
15. sb.append(line).append("\n");
16. }
17. String result = sb.toString();
18. System.out.println(result);
19. }
20. **catch** (Exception e) {
21. e.printStackTrace();
22. }
23. }
24. }

其实就是一个Process类进行调用，然后把shell的执行结果输出到控制台下。

需要注意的是，在调用时需要执行waitFor()函数，因为shell进程是JAVA进程的子进程，JAVA作为父进程需要等待子进程执行完毕。

另外在eclipse控制台输出时并不是边执行边输出，而是shell全部执行完毕后输出，所以如果执行较为复杂的shell脚本看到没有输出时可能会误以为没有执行，这个时候看看终端里面的进程，TOP命令一下就能看到其实shell脚本已经开始执行了。